Markpainting: Adversarial Machine Learning meets Inpainting

David Khachaturov [*] [1] Ilia Shumailov [*] [1] [2] Yiren Zhao [1] Nicolas Papernot [2] Ross Anderson [1]

Abstract

Inpainting is a learned interpolation technique that is based on generative modeling and used to populate masked or missing pieces in an image; it has wide applications in picture editing and retouching. Recently, inpainting started being used for watermark removal, raising concerns. In this paper we study how to manipulate it using our *markpainting* technique. First, we show how an image owner with access to an inpainting model can augment their image in such a way that any attempt to edit it using that model will add arbitrary visible information. We find that we can target multiple different models simultaneously with our technique. This can be designed to reconstitute a watermark if the editor had been trying to remove it. Second, we show that our markpainting technique is transferable to models that have different architectures or were trained on different datasets, so watermarks created using it are difficult for adversaries to remove. Markpainting is novel and can be used as a manipulation alarm that becomes visible in the event of inpainting. Source code is available at: `https://github.com/iliaishacked/markpainting`.

1. Introduction

Improvements to machine learning (ML) have enabled automatic content creation (Ramesh et al., 2021) and manipulation (Yu et al., 2018): a user just needs to provide an image and describe the changes they want as the input to a generative model (Goodfellow, 2016; Korshunova et al., 2017; Antic, 2018). Computer graphics tools brought us digital *inpainting*: programs such as Photoshop enable manipulation of digital images with powerful software and, more recently, ML support (Vincent, 2020). Modern inpainting software lets the user select a patch to be filled in; it then fills this area in with artificially generated content.

One increasingly popular application of inpainting is the removal of objects from photographs. This can be done for malicious purposes. For example, many images are distributed with a watermark that asserts copyright or carries a marketing message; people wishing to reuse the image without permission may want to remove the mark and restore a plausible background in its place. This naturally leads to the question of how we can make watermarks more robust, i.e. difficult to remove. There is substantial literature on using classic signal-processing techniques for mark removal, e.g. from Cox et al. (2007), but such tricks predate recent advances in ML and inpainting more specifically.

In this paper we investigate whether ML inpainters can be manipulated using techniques adapted from the field of adversarial machine learning. Our technique, which we dub *markpainting*, allows for arbitrary manipulation of how inpainters fill in the patch defined by a given 2-bit image *mask*. We do this by setting an arbitrary target image which we wish to appear in the filled-in area. We then generate *perturbations* – small pixel-wise augmentations – which are applied to the original image to manipulate the inpainting algorithms into producing something resembling our target. For example, in Figure 1a, the original image is a black-and-white cartoon; we set the target image to be the same cartoon but with *La Gioconda* pasted onto the otherwise blank canvas. After the application of our technique, the perturbations to the original image ensure that the resulting infilled patch does indeed resemble our target.

We find that the introduction of minor perturbations to input images can force many inpainters to generate arbitrary patches — even patterns not present in the training datasets. Consequently, setting the target to be the original image and applying our markpainting technique makes the image robust against watermark removal as shown in Figure 2. The original (left-most) image has an unsightly watermark that was removed successfully in the middle image by an inpainter. However, after treating the original image with our markpainting technique – setting the target to be the original image itself, to preserve the watermark – the attempt to paint out the watermark fails.

Figure 1b demonstrates the effect of markpainting on six different inpainters. The resulting markpainted sample (bottom

[*]Equal contribution [1]Computer Laboratory, University of Cambridge [2]University of Toronto and Vector Institute. Correspondence to: Ilia Shumailov <ilia.shumailov@cl.cam.ac.uk>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

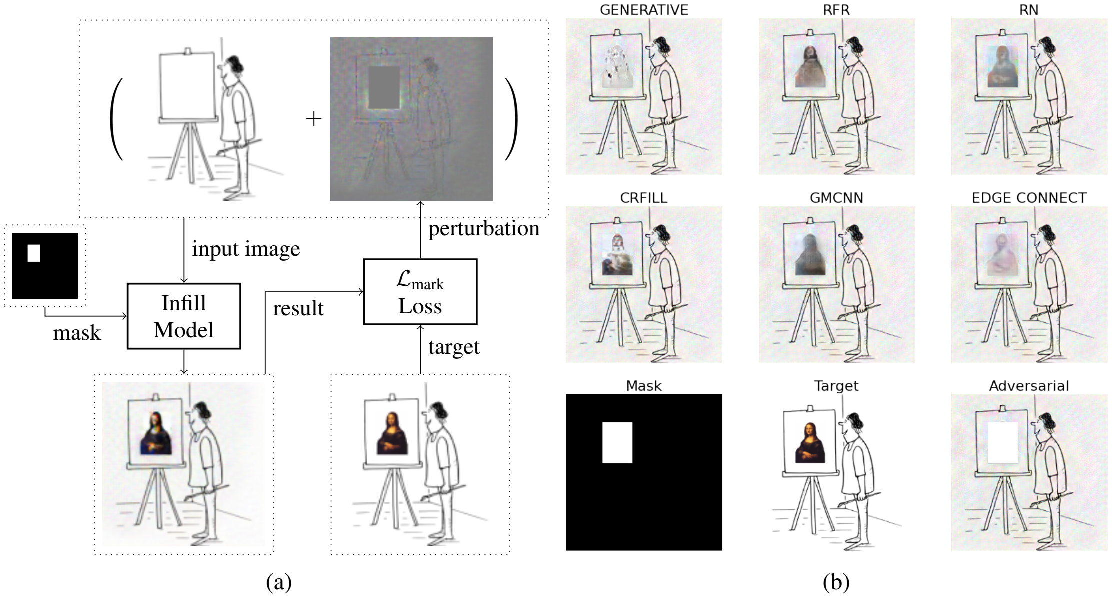


Figure 1. Demonstration of the proposed markpainting technique. The target image is set to be Leonardo da Vinci's *La Gioconda* pasted onto the otherwise-blank cartoon canvas. Figure 1a shows a visual abstract of the proposed markpainting technique, using the CRFILL model. Figure 1b shows the application of markpainting to multiple different inpainting models simultaneously — our technique can target multiple models at once and is not limited to just a single model. The *Adversarial* pane shows the combination of the original input image and the resulting perturbations. The top six images show the result of various inpainters filling-in the rectangular patch on the canvas as defined by the mask. Note that all six inpainters use the same input, namely *Adversarial*. Original cartoon from freesvg.

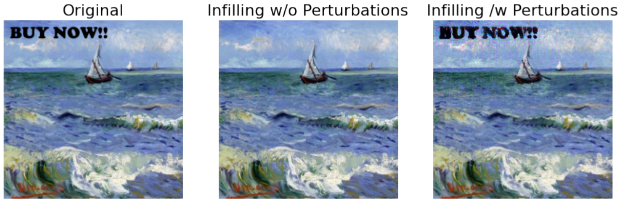


Figure 2. Example of countering watermark removal using markpainting on Vincent van Gogh's *Boats at Sea*. The left-most image depicts the original image with the watermark. The middle image is the result of inpainting the mark without any perturbations, resulting is the successful removal of the watermark. The right-most image contains generated perturbations and has been treated with an inpainter for watermark removal; the output simply restores the mark. Performed on the CRFILL inpainting model with $\epsilon = 0.3$.

right in Figure 1b) is a combination of the original image (top left in Figure 1a) and the accumulated perturbations. We can see that *La Gioconda* (the target) appears on the canvases (the patch to fill in as dictated by the mask) of the final inpainted images (top two rows in Figure 1b). These final images are obtained by running the markpainted sample through each of the inpainting models.

We find that markpainting can work even if the colors and structures of the target image are not present in the input image itself or the dataset the model was trained on. We evaluate the extent to which markpainting transfers from one inpainter to another and within the same inpainter trained on different datasets; the impact of perturbation size; and the viability of mask-agnostic markpainting.

Overall, we make the following contributions:

- We show that inpainting can be manipulated to produce arbitrary content, a technique we name *markpainting*.
- We present a mask-agnostic markpainting method that works regardless of the mask used.
- We evaluate the performance of markpainting thoroughly and find that markpainting a specific target is significantly more effective against more advanced inpainters (a 38% reduction in loss to target in the case of a weak Generative model, compared to a 78% reduction in EdgeConnect's case).
- In a robustness test, we show that markpainted samples sometimes transfer within the same inpainter trained on different datasets, and across different inpainters for markpainting with a target.

2. Broader Impact and Motivation

Malicious actors now manipulate public discourse with artificially generated or manipulated images, such as deepfakes (Goodfellow et al., 2014; Zhang et al., 2020). For example, as shown in Figure 3, it takes no special knowledge to remove a participant from a photo of the 6 January 2021 raid on United States Congress; this is not noticeable without inspecting the image closely. This motivating example led us to study the capacity of inpainting tools to remove or replace objects in images.

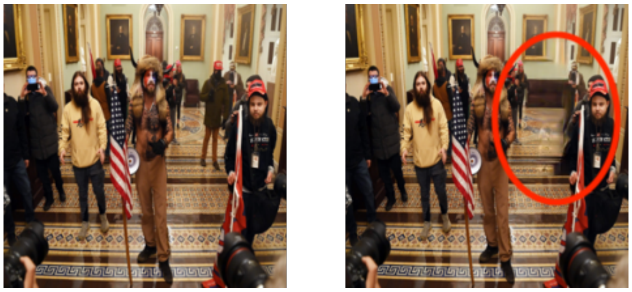


Figure 3. Photo taken from the 6 January 2021 raid on United States Congress. Original on the left; the right photo has been modified using an inpainter to remove a participant. It is near impossible to tell which of the two images is the original one, without closer inspection.

Markpainting can provide protection against evidence tampering by preserving the integrity of published images. Consider an image of a crowd and an attacker who wants to forge evidence by removing a person from the crowd. The defender – e.g. the distributor of the image – does not know which person will be removed, but wants to stop the attacker. If they use our mask-agnostic markpainting technique with a solid color target image (such as pure red), then any attempt to remove a person from the image via inpainting will result in a red patch, clearly marking the image. In practice one would use more subtle techniques, which we discuss later.

3. Related Work

Humans have been restoring paintings for centuries. As ultraviolet light degrades both pigment and bindings in paint, exterior paintwork needs regular reworking; and although artworks kept indoors deteriorate more slowly, they still require upkeep from time to time. Images are touched up for other reasons; after Trotsky fell from favor in Russia, he was airbrushed out of numerous paintings. Digital inpainting is newer, going back to the 1990s when computer-graphics tools started to become both capable and widespread. Early approaches included patch search methods (Bertalmio et al., 2000; Osher et al., 2005) and texture synthesis (Efros & Leung, 1999; Barnes et al., 2009). Those approaches can only work with small missing regions because of the lack of semantic understanding; they are usually computationally expensive because of the time taken to find close matches to missing objects in a large corpus of data (Hays & Efros, 2007).

Recent advances in machine learning have enabled more semantically-aware inpainting. In 2016, Pathak et al. (2016) presented Context Encoders – CNNs trained to predict the contents of an arbitrary image based on its surroundings. They used L2 and an adversarial loss as in generative adversarial networks (Goodfellow et al., 2014). In 2017, Iizuka et al. (2017) built on this work by splitting the discriminator into three: a completion network, a local discriminator and a global discriminator. This architecture allowed inpainting of images of arbitrary size and helped maintain local consistency. Poisson blending allowed further refinement and sharpening of the image. In 2018, Wang et al. (2018) proposed a Generative Multi-column Convolutional Neural Network (GMCNN) with three sub-networks: a generator to inpaint the image, global and local discriminators and a pretrained VGG network for Implicit Diversified Markov Random Fields (ID-MRF) loss calculation. They use filters of different sizes to capture information at different granularity levels, which allowed more fine-grained inpainting. In 2019, Nazeri et al. (2019) used image structure knowledge and developed a two-stage model composed of an edge generator and an image generator.

Recently, there has been significant work in this field. Li et al. (2020b) proposed Recurrent Feature Reasoning (RFR), an inpainting method based on Knowledge Consistent Attention modules. RFR recurrently infers the hole boundaries, then uses them to solve more complex parts of the image. It is split into three parts: an area identification model, a feature reasoning module and a feature-merging operator designed to combine intermediate feature maps. The networks are trained to optimize VGG perceptual and style losses. Li et al. (2020a) proposed a deep generative inpainting network named DeepGin, using a customized ResNet block (He et al., 2016) to allow different receptive fields so that information from both local and distant regions can be gathered efficiently. Jie Yang (2020) built on Nazeri's work with a shared generator to generate both the completed image and its corresponding structures, placing the inpainting problem into a multi-task learning setup. Yu et al. (2020) investigated the feature normalization problem in the context of image inpainting, and proposed a spatially region-wise normalization for image inpainting. Zeng et al. (2020) proposed using a contextually-aware reconstruction loss to replace the contextual attention layers so a network could explicitly borrow from a known region as a reference to inpaint images.

On the adversarial machine learning side of things, in 2013 two separate teams led by Szegedy and Biggio discovered adversarial examples which, during inference, cause a model to output a surprisingly incorrect result (Szegedy et al., 2013; Biggio et al., 2013). In a white-box environment – where the adversary has direct access to the model – such examples

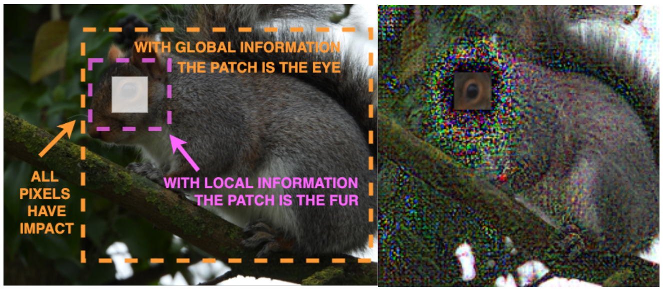


Figure 4. Inpainting of the squirrel eye requires both local and global knowledge. With just local knowledge only the fur patterns could be produced. Image on the right features exaggerated normalized gradients of EdgeConnect (Nazeri et al., 2019) during the first algorithm iteration.

Algorithm 1 General markpainting algorithm

```
**Input:** image $\mathbf{I}$, mask $\mathbf{M}$, target $\mathbf{T}$, perturbation step size $\epsilon'$, iterations $t$, targeted models $\mathbf{\Theta}$
**for** $j = 0$ **to** $t$ **do**
  $\eta \leftarrow \mathbf{0}$
  **for** $\theta \in \mathbf{\Theta}$ **do**
    $\eta \leftarrow \eta + \epsilon' \text{sign}(\nabla_{\mathbf{I}} \mathcal{L}_{\text{mark}}(\theta, \mathbf{I}, \mathbf{T}))$
  **end for**
  $\mathbf{I} \leftarrow \mathbf{I} - (\eta \odot (1 - \mathbf{M}))$
**end for**
$\mathbf{I}_{\text{adv}} \leftarrow \mathbf{I}$
**Output:** markpainted sample $\mathbf{I}_{\text{adv}}$ (combination of original input image $\mathbf{I}$ and the accumulated perturbations)
```

can be found using various gradient-based methods that typically aim to maximize the loss function under a series of constraints (Biggio et al., 2013; Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2019). In a black-box setting, the adversary can transfer adversarial examples from another model (Papernot et al., 2017) or approximate gradients by observing output labels and confidence (Chen et al., 2017). In their various forms, adversarial examples can affect the *confidentiality*, *integrity* and *availability* of machine learning systems (Biggio & Roli, 2018; Papernot et al., 2016; Shumailov et al., 2021).

4. Methodology

4.1. Inpainting

Inpainting fills in information that is missing in an input image. During training, a part of the image is masked out and the inpainter aims to learn how to restore this area.

We define an input RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and a binary mask $\mathbf{M} \in \mathbb{R}^{H \times W}$. The binary mask $\mathbf{M}$ has 0s for the areas to be inpainted and 1s otherwise. We then assume an inpainter f, that populates the region covered by $1 - \mathbf{M}$ taking as input masked input $\hat{\mathbf{I}} = \mathbf{I} \odot (1 - \mathbf{M})$, where $\odot$ represents the Hadamard product. The function f was trained to minimize dissimilarity $\mathcal{L}_{\text{train}}$ between $\hat{\mathbf{I}}$ and $\mathbf{I}$. Training here may involve images of different sizes and irregular masks depending on the system.

4.2. Markpainting

We present two different flavors of markpainting: targeted and mask-agnostic. Targeted markpainting forces the reconstruction to resemble the target image, whilst mask-agnostic markpainting aims to generalize the technique to work with an arbitrary mask. These are presented in Algorithm 1 and Algorithm 2 respectively. Algorithm 1 is visualized in Figure 1a. The formal setup is similar to adversarial example generation (Szegedy et al., 2013; Madry et al., 2019), where the perturbation η is accumulated iteratively from scaled gradients ($\epsilon' \text{sign}(\nabla_{\mathbf{I}} \mathcal{L}_{\text{mark}}(\theta, \mathbf{I}, \mathbf{T}))$).

We define $\mathcal{L}_{\text{mark}}(\theta, x, x') = \mathcal{L}_{\text{network}}(\theta, x) + \alpha l_2(x - x')$, where $\mathcal{L}_{\text{network}}$ is the VGG perceptual loss (Johnson et al., 2016) and l_2 is MSE loss. We use the VGG perceptual loss to measure human visual similarity of markpainting, which is usually missed by pure L2 loss. L2 penalizes large deviations from the target, whilst VGG promotes human-understandable granularity. We set $\alpha = 4$, based on experimentation. The effect of different α values on the markpainted result can be found in Section 4 of our Appendix.

Notice that the perturbation propagated to the natural input is $(\eta \odot (1 - \mathbf{M}))$, because the regions to be infilled are masked out and do not receive gradients.

The technique aims to find a perturbation η with a given perturbation budget ϵ such that the used dissimilarity function $\mathcal{L}_{\text{mark}}$ parameterized by θ is minimized.

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & \mathcal{L}_{\text{mark}}(\theta, f((\mathbf{I} + \eta) \odot (1 - \mathbf{M})), \hat{\mathbf{I}}) \\ \text{subject to} \quad & ||\eta||_p < \epsilon \end{aligned}$$

$||\eta||_p$ is the l_p norm of η and in this paper we use $p = \infty$.

We represent the original input image using $\mathbf{I}$, the original image with our carefully crafted perturbation as $\mathbf{I}_{\text{pert}}$, the naturally inpainted image using $\mathbf{I}_{\text{benign}}$, and the inpainted results of $\mathbf{I}_{\text{pert}}$ as $\mathbf{I}_{\text{mark}}$. We denote the target image using $\mathbf{T}$, and the mask is represented using $\mathbf{M}$.

We find that we can apply our technique to a collection of models $\mathbf{\Theta}$ simultaneously using a single input image $\mathbf{I}$ as detailed in Algorithm 1. An example result of application of markpainting to multiple models simultaneously is presented in Figure 1b and in Section 1 of our Appendix, where the *same* markpainted sample produces a visually-recognizable face, similar to the target, after being

Algorithm 2 EoT markpainting algorithm

```
Input: image $\mathbf{I}$, target $\mathbf{T}$, number of masks $n$, mask size range $[m_{\text{min}}, m_{\text{max}}]$, perturbation step size $\epsilon'$, iterations $t$, targeted models $\mathbf{\Theta}$
Initialize set $\hat{\mathbf{M}}$ to contain $n$ random rectangular masks of size $s \in [m_{\text{min}}, m_{\text{max}}]$
Initialize $\mathbf{M} \leftarrow \varnothing$
for $j = 0$ to $t$ do
    $\mathbf{M} \leftarrow \hat{\mathbf{M}}_i$ for a random $0 \leq i < n$
    $\eta \leftarrow \mathbf{0}$
    for $\theta \in \mathbf{\Theta}$ do
        $\eta \leftarrow \eta + \epsilon' \mathbf{U}(0, 1) \text{sign}(\nabla_{\mathbf{I}} \mathcal{L}_{\text{mark}}(\theta, \mathbf{I}, \mathbf{T}))$
    end for
    $\mathbf{I} \leftarrow \mathbf{I} - (\eta \odot \mathbf{M})$
end for
$\mathbf{I}_{\text{adv}} \leftarrow \mathbf{I}$
Output: markpainted sample $\mathbf{I}_{\text{adv}}$ (combination of original input image $\mathbf{I}$ and the accumulated perturbations)
```

run through six different inpainters.

4.3. Mask-agnostic Markpainting

Although Algorithm 1 works well against a known mask $\mathbf{M}$, there are other cases where we do not know which parts of an image might be tampered with. We adapt our technique to generate an image that will cause a system to markpaint regardless of the mask used. This problem is related to the construction of adversarial examples that work in physical environments under different conditions of lighting and viewing angles. We therefore extend an approach first introduced by Athalye et al. (2018) called *Expectation over Transformation* (EoT).

This extension is presented in Algorithm 2. For this technique, a set of random masks is produced with a given size range $[m_{\text{min}}, m_{\text{max}}]$. We iteratively sample a single mask from the set and apply an algorithm similar to Algorithm 1. We find that further adding stochasticity helps to transfer to unseen masks: we weight the gradient step with a random uniformly-distributed vector $\mathbf{U}(0, 1)$.

4.4. Why does it work?

Inpainting is a complex task, with neural networks trained to manipulate images of arbitrary size and with arbitrary patches. Furthermore, modern inpainters can fill irregular holes. As they are trying to be semantically aware and display both local and global consistency, they need to understand the global scenery well. That in turn makes them dependent not only on the area around the patch, but on the whole image. Imagine trying to fill in a hole around the squirrel eye depicted in Figure 4. Here, local information (shown in pink) would suggest that it has to be filled with fur. Global information (shown in orange) on the other hand, should tell the inpainter that the picture features a squirrel in a particular pose and that an eye should be located there. As illustrated in the gradient visualization in Figure 4, gradients focus on both the area around the eye and the rest of the image. This dependency on global information makes inpainting both complex and prone to manipulation. The markpainter does not need to concentrate their perturbation around the patch area but can scatter it all over the image.

While at first glance markpainting seems similar to older techniques, such as ones proposed by Levin et al. (2004), there are fundamental differences in the two approaches. Inpainting requires a semantic understanding of the scenery and heavily depends on global information, as shown in Figure 4. Furthermore, markpainting can produce artifacts that are semantically meaningless for the model and not present in its training distribution.

5. Evaluation

In this section, we evaluate the performance of targeted markpainting in Section 5.2, and the effect of different masks and target images in Section 5.3. Section 5.4 focuses on the transferability of the generated samples, while Section 5.5 discusses mask-agnostic markpainting.

5.1. Datasets and Models

In Table 1 we list the inpainter systems used in the evaluation. Our evaluation covers systems that provide different levels of granularity of the inpainted regions and different levels of representation. We intentionally chose a variety of inpainters with differing levels of performance and from different years. We use pretrained models provided by the authors of the respective systems. Table 1 also indicates the datasets with which these inpainters are pretrained. A maximum perturbation budget of ϵ was used with a step size of $\epsilon' = \frac{\epsilon}{50}$ unless specified otherwise. We justify the parameter choices in Section 4 of the Appendix. We clip the markpainted image at each iteration to make sure that the total perturbation budget does not exceed ϵ.

Table 1. Inpainters used in the evaluation

System		**Dataset**
Generative	(Yu et al., 2018)	ImageNet (Deng et al., 2009)
GMCNN	(Wang et al., 2018)	CelebA-HQ (Liu et al., 2015)
EdgeConnect	(Nazeri et al., 2019)	Paris StreetView (Doersch et al., 2012), CelebA (Liu et al., 2015), Places2 (Zhou et al., 2017)
RFR	(Li et al., 2020b)	Paris StreetView, CelebA
RN	(Yu et al., 2020)	Places2
CRFILL	(Zeng et al., 2020)	Places2, Salient Object Segmentation (Xiong et al., 2019)

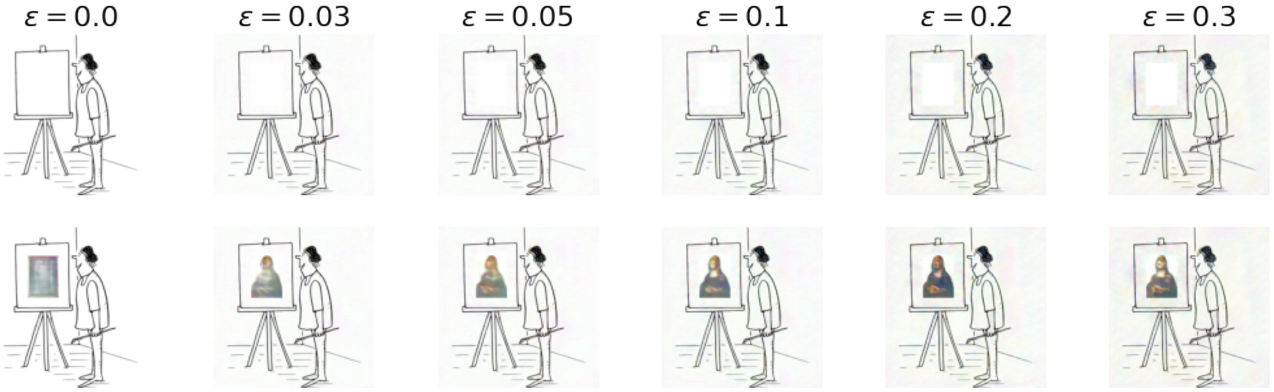


Figure 5. Inpainting with an increasing perturbation budget. Top row is the perturbed images generated using markpainting, and second row is the inpainted results of these perturbed images. We target the RN inpainter with 500 iterations and a step size of $\epsilon/100$. Note that this example is really hard, because we are filling a black and white image with color. Details are discussed in Section 5.2.

The systems are evaluated on *places_subset16*, a series of 16 randomly-selected images from the Places2 dataset (Zhou et al., 2017)[1] – using fixed random masks of three different sizes respectively covering 5%, 10% and 20% of the image. We use three solid-color targets for evaluation: pure red, green, and blue. Further details are provided in the Appendix.

5.2. Targeted Markpainting

Figure 5 illustrates the visual effect of applying markpainting to the inpainter based on Region Normalization (RN) (Yu et al., 2020) with an increasing perturbation budget. The top row shows the markpainted images we produced and the second row shows the final inpainted results. The inpainting task here is complex, as it requires constructing a colored patch from a black-and-white image. Even with a small budget $\epsilon = 0.05$ that is barely perceptible, RN markpaints the region with a lot of detail from the target image: we can see the structure and edges of *La Gioconda*. At larger ϵ values, facial details start to appear.

Table 2 presents an evaluation on the *places_subset16* dataset, averaging results from all possible input-mask-target combinations. It is seen that larger perturbation budgets ϵ cause the dissimilarity metric $\mathcal{L}_{\text{mark}}$ and l_2 norm distance to reduce, and PSNR and SSIM to increase, between the markpainted image $\mathbf{I}_{\text{mark}}$and the target $\mathbf{T}$. This is expected since the increased budget results in better reconstruction of the target, with the effect that the samples lose resemblance to the benign reconstruction $\mathbf{I}_{\text{benign}}$. Larger budgets help fine-grained artifacts from the target to appear in $\mathbf{I}_{\text{mark}}$, whilst sacrificing imperceptibility. We empirically see that $\epsilon = 0.05$ is usually invisible, while allowing for a good level of reconstruction detail.

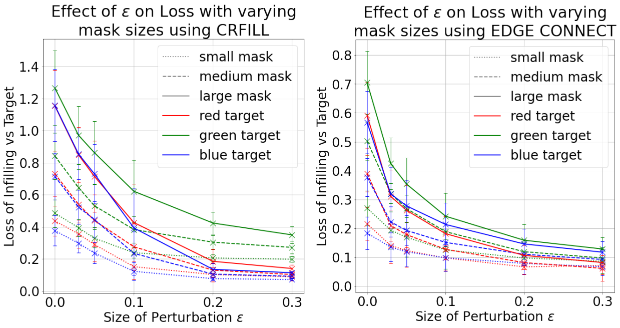


Figure 6. The impact that perturbation sizes have on $\mathcal{L}_{\text{mark}}$ between the markpainted patch and the target (lower is better). We run markpainting over 100 iterations. The x-axis shows different perturbation budgets and the vertical axis is the loss between $\mathbf{I}_{\text{mark}}$ and $\mathbf{T}$. Results are averaged across the *places_subset16* images, with $\pm\sigma$ error bars. Details are discussed in Section 5.3.

The effectiveness of our technique when targeting multiple models simultaneously is discussed in Section 1 of our Appendix.

5.3. Impact of Mask-target Choice

To illustrate the effectiveness of our markpainting technique, we show how it performs under varying mask sizes and target images in Figure 6. We find that although mask size has an influence on technique performance, the color of the target image has a greater impact. We find that green color areas are harder to markpaint for both models, whereas blues are easiest. We suspect this is due to the source images having little green, and the training datasets perhaps also lacking this color.

[1] We abstain from evaluating on CelebA dataset due to ethical concerns over the dataset labels.

Distance to:		Original				Adversarial Target				Benign			
ϵ		Loss	l_2	PSNR	SSIM	Loss	l_2	PSNR	SSIM	Loss	l_2	PSNR	SSIM
0.0	GENERATIVE	0.467	0.111	12.491	0.250	0.755	1.186	-0.622	0.036	**0.179**	0.000	134.254	1.000
	RFR	0.279	0.027	19.156	0.387	0.433	0.292	5.464	0.090	**0.001**	0.000	144.880	1.000
	RN	0.300	0.025	17.891	0.420	0.473	0.292	5.438	0.104	**0.001**	0.000	inf	1.000
	CRFILL	0.470	0.109	13.342	0.319	0.796	1.205	-0.683	0.044	**0.180**	0.000	inf	1.000
	GMCNN	0.485	0.122	10.891	0.210	0.721	1.136	-0.432	0.047	**0.181**	0.000	inf	1.000
	EDGE CONNECT	0.290	0.025	18.198	0.390	0.422	0.299	5.383	0.104	**0.001**	0.000	135.133	1.000
0.05	GENERATIVE	0.441	0.108	12.121	0.225	0.623	1.093	-0.281	0.050	**0.289**	0.028	17.483	0.476
	RFR	0.332	0.040	15.533	0.325	0.310	0.232	6.547	0.139	**0.161**	0.019	18.374	0.630
	RN	0.386	0.081	11.697	0.266	**0.203**	0.153	8.763	0.239	0.271	0.062	13.094	0.435
	CRFILL	0.491	0.244	7.291	0.128	0.509	0.782	1.481	0.135	**0.406**	0.186	8.676	0.232
	GMCNN	0.605	0.818	3.047	-0.018	0.579	1.349	-0.453	0.040	**0.432**	0.698	4.491	0.038
	EDGE CONNECT	0.351	0.057	13.032	0.317	0.216	0.206	7.120	0.214	**0.190**	0.042	14.655	0.546
0.3	GENERATIVE	0.489	0.184	8.676	0.132	0.466	0.768	1.284	0.140	**0.379**	0.135	9.746	0.167
	RFR	0.412	0.095	10.965	0.288	**0.143**	0.110	10.191	0.247	0.278	0.079	11.604	0.381
	RN	0.456	0.154	8.620	0.216	**0.067**	0.049	14.125	0.322	0.365	0.140	8.970	0.315
	CRFILL	0.698	0.896	0.759	0.035	**0.159**	0.060	14.076	0.695	0.646	0.884	0.789	0.047
	GMCNN	0.681	1.112	0.968	-0.067	0.533	1.457	0.175	0.081	**0.511**	0.993	1.851	-0.066
	EDGE CONNECT	0.428	0.112	10.336	0.285	**0.089**	0.091	11.166	0.268	0.289	0.103	10.865	0.398

Table 2. Impact of markpainting on different inpainter models. This table reports the loss ($\mathcal{L}_{\text{mark}}$ from Section 4.2), L2 norms, peak signal to noise ratio (PSNR) and structural index similarity (SSIM) for assessing the inpainted image quality. Markpainting is applied to each individual inpainter and evaluated on the same inpainter with different perturbations budgets; this table is a compact version of Table 1 in our Appendix, where more epsilon values are available. In this table we highlight cases where the loss to the target image is better than to the original reconstruction. For these three meta-columns, we report how the markpainted patch ($\mathbf{I}_{\text{mark}} \odot \mathbf{M}$) compares to different images in different metrics. 'Original' refers to the original image patch ($\mathbf{I} \odot \mathbf{M}$), 'Adversarial target' is the target image used ($\mathbf{T} \odot \mathbf{M}$), and 'Benign' is the image that the model would have produced without any adversarial perturbation ($\mathbf{I}_{\text{benign}} \odot \mathbf{M}$). Increasing the perturbation budget ϵ increases the similarity between the markpainted patch and the target but decreases similarity to the original image and benign inpainted patch. Details are discussed in Section 5.2.

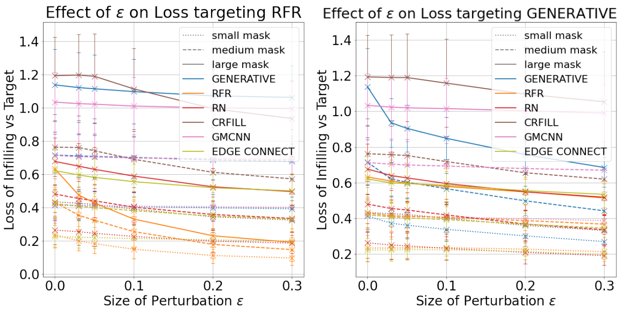
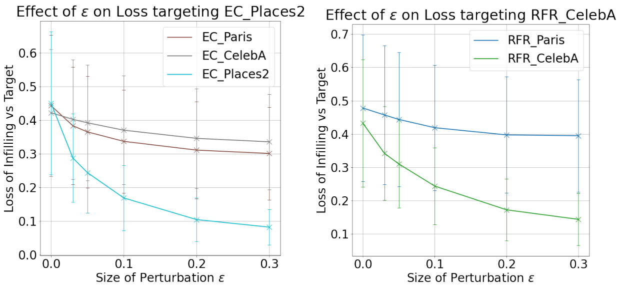


Figure 7. Showing technique transferability between different models using $\mathcal{L}_{\text{mark}}$ between the markpainted patch and the target. The perturbations come from targeting the model listed in the title but the errors are shown for the 6 color-coded models. Results are averaged across all possible input-mask-target combinations, with $\pm\sigma$ error bars to highlight the standard deviation in obtained results. Details are described in Section 5.4.



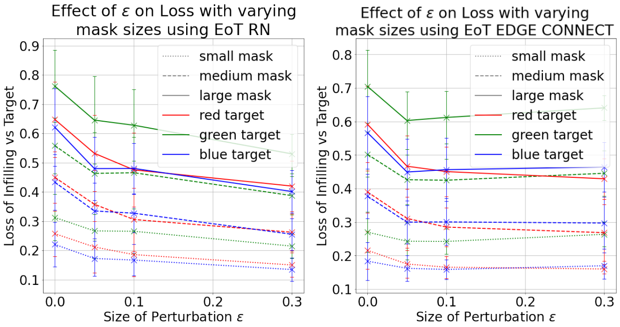
(a) EdgeConnect, trained on Places2.

(b) RFR, trained on CelebA.

Figure 8. Showing technique transferability between the same model, trained on different datasets, using $\mathcal{L}_{\text{mark}}$ between the markpainted patch and the target. Results are averaged across all possible input-mask-target combinations. Notice the large $\pm\sigma$ error bars, indicating high variability in transferability depending on the input combination.

5.4. Transferability of Targeted Markpainted Examples

Here we investigate the transferability of markpainted images in a blind black-box scenario, using previously-constructed markpainted samples to fool other models without knowledge of their internals. We investigate transferability across both model architectures and datasets. We report mean model performance with an increasing perturbation budget. Although an increased budget helps with transferability, the improvement is marginal in most cases. Note that variances of the measurements here are large; they



(a) EoT technique on RN.

(b) EoT technique on EdgeConnect.

Figure 9. Effect of ϵ on the effectiveness of the proposed mask-agnostic markpainting technique. Results are averaged across the *places_subset16* images, with $\pm\sigma$ error bars. It is evident that the technique's effectiveness is very much architecture dependent, with a high sensitivity to the input.

reflect differences in input images and how different color targets transfer across models.

In Figure 7, we turn to the question of whether inpainter models show greater transferability if pretrained with the same dataset. RN and CRFILL, both trained on Places2, demonstrate a correlated decreasing pattern on the right plot of Figure 7, showing that inpainters trained on the same dataset might suffer from transferred markpainting samples.

In Figure 8 we demonstrate the transferability of markpainted examples within the same model architecture trained on different datasets. For this experiment we use EdgeConnect – trained on CelebA, Paris StreetView, and Places2 – and RFR – trained on CelebA and Paris StreetView. Effectiveness of markpainted examples degrades to varying degrees when used by a model trained on a different dataset. However, the graphs demonstrate that markpainted examples are transferable within the same model architecture.

In general, we found that transferability exists across different inpainter models and datasets. This is broadly equivalent to the robustness of a watermark protected by our technique. It shows greater transferability when inpainters are trained on the same dataset or share the model architecture.

5.5. Mask-agnostic Markpainting

Figure 9 shows the effectiveness of our EoT method for mask-agnostic markpainting. The number of iterations was taken to be 1500 with a step size of $\frac{\epsilon}{30}$, with $m_{\text{min}} = 0.01$ and $m_{\text{max}} = 0.1$. The evaluation masks are taken to be fixed random masks covering 2.5%, 5% and 10% of the image. The effectiveness was found to be architecture dependent. Moreover, certain images were more susceptible to markpainting than others. Investigating the exact causes of this architecture and image dependence is left to future work.

6. Discussion

6.1. Countering Markpainting

We have shown that modern inpainters can be manipulated to inpaint arbitrary target images. This naturally leads to the question of how one can counter markpainting. As markpainting aims to be explicitly imperceptible, it usually does not disrupt lower parts of the frequency spectra responsible for sharp edges, instead concentrating on the higher-frequency components. We propose a mechanism which accounts for this.

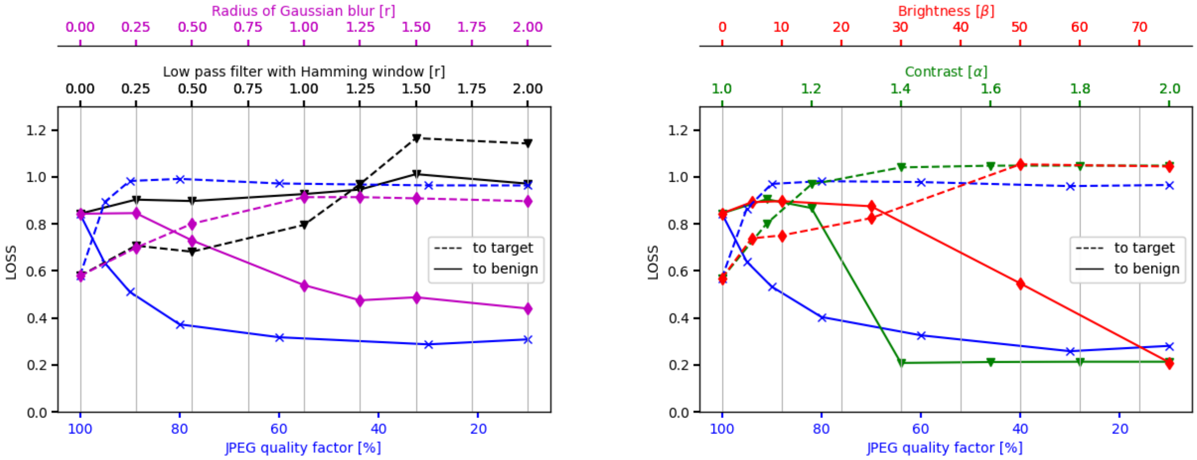
We find that transformation-based manipulations work relatively well in countering markpainting. Figure 10 shows dissimilarity $\mathcal{L}_{\text{mark}}$ between the markpainted patch and the target/benign images. We test five different transformations: JPEG compression, low-pass filtering, Gaussian blurring, contrast adjustments, and brightness adjustments. Each manipulation significantly reduced markpainting performance, but had different impact on the inpainting performance in the benign cases. Simple low-pass filtering reduces similarity of the reconstruction to the target image, but also causes significant deviation from the benign reconstruction. This highlights the trade-off between countering markpainting and preserving the benign inpainted patch. Although some transformations decrease the performance of markpainting, they change the original image significantly as well. Thus, manual human involvement appears to be required, which is highly likely to limit the scalability of abuse based on inpainting.

6.2. Interpretability of Markpainting

Unlike adversarial examples for classification tasks, markpainting can be interpreted. Indeed, we find that we could often visually tell what an increased perturbation budget was changing in our perception of the inpainter model. Evaluation suggests that although markpainting can be made transferable, it usually is not. We find that it is, perhaps intuitively, harder to markpaint colors or shapes that are not present in the original image. Complex shapes, and contours that do not naturally extend from the mask's boundaries, also prove to be a challenge. In contrast, models that have been trained on the CelebA dataset are easier to fool into markpainting faces, as demonstrated strikingly in a visualization provided in Section 1 of our Appendix.

7. Conclusion

We introduce the idea of *markpainting*: fooling an inpainting system into generating a patch similar to an arbitrary target. Moreover, we demonstrate through mask-agnostic markpainting that the technique does not need to be restricted to a particular mask to be effective. We also show the existence of some degree of transferability of these ad-



(a) JPEG compression, low pass filtering and Gaussian blur.

(b) JPEG compression, brightness and contrast.

Figure 10. Effect of different transformation-based defenses. Results are from performing these transformations on the watermark example presented in Figure 2.

versarial examples both within a single model and between different model architectures.

Markpainting has wide implications. Image owners can now protect their digital assets with less removable watermarks, or treat them so that any later manipulation such as object removal becomes easier to detect.

Acknowledgments

We thank the reviewers for their insightful feedback. We want to explicitly thank Mohammad Yaghini, Stephan Rabanser, Gabriel Deza, Natalie Dullerud, Ali Shahin Shamsabadi and Nicholas Boucher for their help and comments. This work was supported by CIFAR (through a Canada CIFAR AI Chair), by EPSRC, by Apple, by Bosch Forschungsstiftung im Stifterverband, by NSERC, and by a gift from Microsoft. We also thank the Vector Institute's sponsors.

References

Antic, J. Deoldify. https://github.com/jantic/DeOldify/, 2018.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples, 2018.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pp. 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085. doi: 10.1145/344779.344972. URL https://doi.org/10.1145/344779.344972.

Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. A. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):101:1–101:9, 2012.

Efros, A. A. and Leung, T. K. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE* *international conference on computer vision*, volume 2, pp. 1033–1038. IEEE, 1999.

Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, 2015.

Hays, J. and Efros, A. A. Scene completion using millions of photographs. *ACM Trans. Graph.*, 26(3): 4–es, July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276382. URL https://doi.org/10.1145/1276377.1276382.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073659. URL https://doi.org/10.1145/3072959.3073659.

Jie Yang, Zhiquan Qi, Y. S. Learning to incorporate structure knowledge for image inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12605–12612, 2020.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution, 2016.

Korshunova, I., Shi, W., Dambre, J., and Theis, L. Fast face-swap using convolutional neural networks, 2017.

Levin, A., Lischinski, D., and Weiss, Y. Colorization using optimization. *ACM Transactions on Graphics*, 23, 06 2004. doi: 10.1145/1015706.1015780.

Li, C.-T., Siu, W.-C., Liu, Z.-S., Wang, L.-W., and Lun, D. P.-K. Deepgin: Deep generative inpainting network for extreme image inpainting, 2020a.

Li, J., Wang, N., Zhang, L., Du, B., and Tao, D. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020b.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2019.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., and Ebrahimi, M. Edgeconnect: Generative image inpainting with adversarial edge learning, 2019.

Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4 (2):460–489, 2005.

Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ramesh, A., Pavlov, M., Goh, G., and Gray, S. Dall·e: Creating images from text, Jan 2021. URL https://openai.com/blog/dall-e/.

Shumailov, I., Zhao, Y., Bates, D., Papernot, N., Mullins, R., and Anderson, R. Sponge examples: Energy-latency attacks on neural networks. In *6th IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Vincent, J. Photoshop's ai neural filters can tweak age and expression with a few clicks, 2020. URL www.theverge.com/2020/10/20/21517616/adobe-photoshop-ai-neural-filters-beta\-launch-machine-learning.

Wang, Y., Tao, X., Qi, X., Shen, X., and Jia, J. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 331–340, 2018.

Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., and Luo, J. Foreground-aware image inpainting, 2019.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., Zhang, Z., and Liu, S. Region normalization for image inpainting. In *AAAI*, pp. 12733–12740, 2020.

Zeng, Y., Lin, Z., Lu, H., and Patel, V. M. Image inpainting with contextual reconstruction loss, 2020.

Zhang, B., Zhou, J. P., Shumailov, I., and Papernot, N. Not my deepfake: Towards plausible deniability for machine-generated media, 2020.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.